COVID-19 DATA ANALYSIS AND FORECASTING

A PROJECT REPORT

Submitted by

REAHAAN SHERIFF I

(2019202045)

submitted to the Faculty of

INFORMATION SCIENCE AND TECHNOLOGY

in partial fulfillment for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY

CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"COVID-19 DATA ANALYSIS AND FORECASTING"** is the bonafide work of **REAHAAN SHERIFF I (2019202045)** who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE: CHENNAI
DATE: 13.06.22

Ms.P.S. APIRAJITHA
TEACHING FELLOW
INTERNAL GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

Dr. S SRIDHAR
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

COVID-19 has sparked a worldwide pandemic, with the number of infected cases and deaths rising on a regular basis. Along with recent advances in soft computing technology, researchers are now actively developing and enhancing different mathematical and machine-learning algorithms to forecast the future trend of this pandemic. Thus, if we can accurately forecast the trend of cases globally, the spread of the pandemic can be controlled. In this project, a LSTM model, Prophet Model and ARIMA model will be used on a time-series dataset to forecast the cases of COVID-19 in future. Machine learning (ML) is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. Long Short Term Memory is a kind of recurrent neural network.

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values.

திட்டப்பணி சுருக்கம்

COVID-19 உலகளாவிய தொற்றுநோயைத் தூண்டியுள்ளது, பாதிக்கப்பட்ட வழக்குகள் மற்றும் இறப்புகளின் எண்ணிக்கை தொடர்ந்து அதிகரித்து வருகிறது. சாஃப்ட் கம்ப்யூட்டிங் தொழில்நுட்பத்தில் சமீபத்திய முன்னேற்றங்களுடன், ஆராய்ச்சியாளர்கள் இப்போது இந்த தொற்றுநோயின் எதிர்கால போக்கை முன்னறிவிப்பதற்காக பல்வேறு கணித மற்றும் இயந்திர கற்றல் வழிமுறைகளை தீவிரமாக உருவாக்கி மேம்படுத்தி வருகின்றனர். எனவே, உலகளவில் வழக்குகளின் போக்கை துல்லியமாக கணிக்க முடிந்தால், தொற்றுநோய் பரவுவதைக் கட்டுப்படுத்த முடியும். இந்தத் திட்டத்தில், எதிர்காலத்தில் COVID-19 பாதிப்புகளை முன்னறிவிப்பதற்காக, LSTM மாதிரி, Prophet மாதிரி மற்றும் ARIMA மாதிரி ஆகியவை நேர-தொடர் தரவுத்தொகுப்பில் பயன்படுத்தப்படும். இயந்திர கற்றல் என்பது ஒரு வழிமுறையின் ஒரு வகையாகும், இது மென்பொருள் பயன்பாடுகள் வெளிப்படையாக திட்டமிடப்படாமல் விளைவுகளை கணிப்பதில் மிகவும் துல்லியமாக இருக்க அனுமதிக்கிறது. இயந்திரக் கற்றலின் அடிப்படையானது, உள்ளீட்டுத் தரவைப் பெறக்கூடிய வழிமுறைகளை உருவாக்குவது மற்றும் புதிய தரவு கிடைக்கும்போது வெளியீடுகளைப் புதுப்பிக்கும் போது வெளியீட்டைக் கணிக்க புள்ளியியல் பகுப்பாய்வைப் பயன்படுத்துவதாகும். நீண்ட குறுகிய கால நினைவகம் என்பது ஒரு வகையான தொடர்ச்சியான நரம்பியல் நெட்வொர்க் ஆகும்.

Prophet என்பது ஒரு சேர்க்கை மாதிரியின் அடிப்படையில் நேரத் தொடர் தரவை முன்னறிவிப்பதற்கான ஒரு செயல்முறையாகும், இதில் நேரியல் அல்லாத போக்குகள் வருடாந்திர, வாராந்திர மற்றும் தினசரி பருவநிலை மற்றும் விடுமுறை விளைவுகளுடன் பொருந்துகின்றன. வலுவான பருவகால விளைவுகள் மற்றும் பல பருவகால வரலாற்றுத் தரவுகளைக் கொண்ட நேரத் தொடரில் இது சிறப்பாகச் செயல்படுகிறது. தொலைந்து போன தரவு மற்றும் போக்கில் மாற்றங்கள் ஆகியவற்றில் நபி வலிமையானவர், மேலும் பொதுவாக வெளியாட்களை நன்கு கையாளுகிறார். ஒரு தன்னியக்க ஒருங்கிணைக்கப்பட்ட நகரும் சராசரி அல்லது ARIMA என்பது ஒரு புள்ளிவிவர பகுப்பாய்வு மாதிரியாகும், இது தரவுத் தொகுப்பை நன்கு புரிந்துகொள்ள அல்லது எதிர்கால போக்குகளைக் கணிக்க நேரத் தொடர் தரவைப் பயன்படுத்துகிறது. ஒரு புள்ளியியல் மாதிரியானது கடந்த கால மதிப்புகளின் அடிப்படையில் எதிர்கால மதிப்புகளை முன்னறிவித்தால் அது தானாகவே பின்னடைவு ஆகும்

ACKNOWLEDGEMENT

Its my honor to communicate my genuine thanks to my venture guide **Ms.P.S. Apirajitha**, Teaching Fellow, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, Chennai for her distinct fascination, rousing direction, steady consolation and backing with my work during every one of the stages, to bring this postulation into realization.

I would like to express my sincere thanks to the project committee members, **Dr.Saswati Mukherjee**, Professor, **Dr.M.Vijayalakshmi**, Associate Professor,**Dr.E.Uma**, Assistant Professor, **Ms.P.S.Apirajitha**, Teaching Fellow, **Ms.C.M.Sowmya**, Teaching Fellow Department of Information Science and Technology, Anna University,Chennai for giving their important ideas, support and steady inspiration all through the length of my task.

REAHAAN SHERIFF I

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

LSTM	Long Short Term Memory
ARIMA	Auto Regressive Integrated Moving Average
COVID	Corona Virus Disease
ML	Machine Learning
HTML	HyperText Markup Language
CSS	Cascading Style Sheets

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Machine learning (ML) is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long-term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give an efficient performance. Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well. An autoregressive integrated moving average, or ARIMA, is a statistical analysis model that uses time series data to either better understand the data set or to predict future trends. A statistical model is autoregressive if it predicts future values based on past values.

1.2 OBJECTIVE

To predict the cases and peak case date for COVID-19 in India as

accurately as possible. I will be using LSTM, Prophet, ARIMA and Random Forest Regressor machine learning models.

1.3 TECHNOLOGIES USED

Google COLAB used to train the model using machine learning algorithms and to interact with the machine learning model a front end user interface is build using HTML, CSS, BOOTSTRAP and integrated the machine learning model with backend server DJANGO.

1.3.1 Machine Learning Libraries

The hardware requirements mentioned below mainly describe the minimum hardware requirements for the application to run. But is always advisable to deploy the application at the hardware device with the recommended specifications.

- Pandas: It is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.
- Numpy: It offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more. Interoperable.
- Sklearn: Scikit-learn is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

- Tensorflow: It is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.
- Matplotlib: It is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

1.3.2 Frontend Tools

The hardware requirements mentioned below mainly describe the minimum hardware requirements for the application to run. But is always advisable to deploy the application at the hardware device with the recommended specifications.

- HTML: The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.
- CSS: Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.
- Bootstrap: Bootstrap is a potent front-end framework used to create modern websites and web apps. It's open-source and free to use, yet features numerous HTML and CSS templates for UI interface

elements such as buttons and forms. Bootstrap also supports JavaScript extensions.

- Django: Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.
- VS Code: Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

1.4 MACHINE LEARNING MODELS

1.4.1 Long-Short Term Memory

LSTM is a novel recurrent network architecture in combination with a proper gradient-based learning algorithm. LSTM is designed to overcome these error back-flow problems. It can learn to bridge time intervals over 1000 steps, even in the noisy case, compact input sequences, without the loss of short time delay capabilities. An efficient, gradient-based algorithm achieves this for an architecture enforcing constant error flow that is neither exploding nor disappearing through internal states of each unit.

In principle, an LSTMs can use its memory cells to remember long-range information and track the various attributes of text it is currently

processing. For instance, it is a simple exercise to write gadget cell weights that would allow the cell to keep track of whether it is inside a quoted string.

During the preparation cycle of an organization, the primary goal is to limit misfortune (as far as blunder or cost) saw in the yield when preparing information is sent through it. We figure the inclination, that is, misfortune concerning a specific arrangement of loads, change the loads as needs be, and rehash this cycle until we get an ideal arrangement of loads for which misfortune is least. This is the idea of backtracking. At times, it so happens that the slope is practically immaterial. It must be noticed that the slope of a layer relies upon specific segments in the progressive layers. On the off-chance that a portion of these segments is little (under 1), the outcome acquired, which is the inclination, will be much more modest. This is known as the scaling impact. When this angle is duplicated with the learning rate, which is a little worth running between 0.1 and 0.001, it brings about a more modest worth. As an outcome, the modification in loads is minuscule, delivering nearly a similar yield as in the past. Likewise, if the inclinations are huge in esteem because of the enormous estimations of parts, the loads get refreshed to an incentive past the ideal worth. This is known as the issue of detonating inclinations. To evade this scaling impact, the neural organization unit was re-underlying so that the scaling factor was fixed to one. The phone was then enhanced by a few gating units and was called LSTM.

1.4.2 Facebook Prophet Model

The prophet is a procedure for forecasting time series data based upon the idea of additive modelling where non-linear trends fit with annual, week-to-week, and day-to-day seasonality. The data here is expected to have good seasonal effects and should have various seasons (or) periods of past historical data. Prophet can deal with missing data and changes in the trend. It

can also manage outliers well. It is precise and fast, and its various applications for generating reliable forecasts. The prophet is fully automated, which helps in getting a sensible forecast on messy data without manual effort. It additionally includes methods in which users can tweak and adjust forecasts. By using Human –Interpretable parameters, the forecast can be improved [13]. The prophet is available in Python and R, and they use the same Stan code for fitting the model. The prophet is an additive regression model with significant elements such as a piecewise linear or logistic growth curve trend. It instantaneously finds changes in patterns by selecting change points from the data. An annual seasonal component modeled using the Fourier series [14]. Figure 2 shows the Prophet components plot, which tells about the model it has fit.

The above Figure 2 shows the Prophet components plot, which tells about the model it has fit. The components plot comprises a weekly component and the monthly component of the model, shown using the curves in the plot. Prophet offers unpredictability intervals for the trend component by mimicking future trend changes to the time series. Prophet uses Stan's probabilistic programming language to execute the core of the procedure. Stan carries out the MAP optimization for parameters very swiftly in less than one second and also gives us a choice to approximate parameter uncertainty by making use of the Hamiltonian Monte Carlo algorithm, and enables us to re-use the fitting procedure throughout numerous interface languages. Prophet is an additive regression model y(t) and the equation for the prediction model goes like this y(t)=g(t)+s(t)+h(t)+t.

1.4.3 Auto Regressive Integrated Moving Average Model

Auto-Regressive Integrated Moving Average (ARIMA), is a time-series auto-regressive technique that calculates future short-term predictions from analyzing time-series of historical data. ARIMA has been used

in the past to predict several disease outbreaks such as Hemorrhagic Fever with Renal Syndrome (HFRS), Hand–Foot–Mouth Disease (HFMD), Hepatitis-B, as well as the recently emerged COVID-19 virus.

Since the emergence of COVID-19 in late 2019, researchers have been studying the pattern and rate of its infection using different mathematical modeling methodologies popularized in epidemiology research. A prediction model was developed using Artificial Neural Networks (ANN) to estimate the growth of COVID-19 cases in the world based on geo-location and 2-week past data. This research compared the predicted numbers produced by their model with the actual values and found them to be closely matched. A model called FPASSA-ANFIS is proposed. It improves the adaptive neuro-fuzzy inference system (ANFIS) using an enhanced flower pollination algorithm (FPA) by using the salp swarm algorithm (SSA). Their model's results showed good performance when comparing different accuracy measures against several other existing models. A Neural Network model for COVID-19 spread prediction is proposed. The prediction model learns using NAdam training model and produced predictions for different countries and regions around the world. The proposed model achieved highly accurate results averaging 87.7 percent for most regions.

1.4.4 Random Forest Regression

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a

number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

1.5 ORGANIZATION OF THE REPORT

The thesis is organized into 6 chapters, describing each part of the project with detailed illustration and system design diagrams. The chapters are as follows:

Chapter 2: discusses the related works made on the proposed work, by analyzing issues of the existing system.

Chapter 3: discusses the functionalities of the proposed system and explains about each of the modules in detail.

Chapter 4: illustrates the implementation results of the proposed work

Chapter 5: concludes the report by summarizing the results and proposes possible enhancements that can be extended as the future work.

The previously mentioned five modules are circled back to the references which intentionally makes sense of and list all the reference archives utilized during the different periods of the undertaking, which incorporates the diary papers, meeting papers, white papers, articles and sites alluded for instructional exercises.

CHAPTER 2

LITERATURE SURVEY

2.1 CORONAVIRUS DISEASE (COVID-19) CASES ANALYSIS USING MACHINE-LEARNING APPLICATIONS

Ameer Sardar et al. [1] states that the purpose of this study is to detect the role of machine-learning applications and algorithms in investigating and various purposes that deals with COVID-19. Their findings show that machine learning can produce an important role in COVID-19 investigations, prediction, and discrimination.In conclusion, machine learning can be involved in the health provider programs and plans to assess and triage the COVID-19 cases. Supervised learning showed better results than other Unsupervised learning algorithms by having 92.9 percent testing accuracy. In the future recurrent supervised learning can be utilized for superior accuracy.

2.2 OPTIMIZING LSTM FOR TIME SERIES PREDICTION IN INDIAN STOCK MARKET

Anita Yadav et al.[2] states that Long Short Term Memory (LSTM) is one of the most popular deep learning models. It is applied to time series prediction which is a particularly hard problem to solve due to the presence of long term trend, seasonal and cyclical fluctuations and random noise. The performance of LSTM was highly dependent on choice of several hyper-parameters which need to be chosen very carefully, in order to get good results. Being a relatively new model, there are no established guidelines for configuring LSTM. [2] In this paper this research gap was addressed. A dataset was created from the Indian stock market and an LSTM model was developed

for it. It was then optimized by comparing stateless and stateful models and by tuning for the number of hidden layers.

2.3 A COMPARATIVE STUDY:TIME-SERIES ANALYSIS METHODS FOR PREDICTING COVID-19 CASE TREND

Tolga Ergen et al.[3] states that Since 2019, COVID-19, as a new acute respiratory disease, had struck the whole world, causing millions of death and threatening the economy, politics, and civilization. Therefore, an accurate prediction of the future spread of COVID-19 becomes crucial in such a situation. [3] In this comparative study, four different time-series analysis models, namely the ARIMA model, the Prophet model, the Long Short-Term Memory (LSTM) model, and the Transformer model, are investigated to determine which has the best performance when predicting the future case trends of COVID-19 in six countries. After obtaining the publicly available COVID-19 case data from Johns Hopkins University Center for Systems Science and Engineering database, we conduct repetitive experiments which exploit the data to predict future trends for all models.The performance is then evaluated by mean squared error (MSE) and mean absolute error (MAE) metrics. The results show that overall the LSTM model has the best performance for all countries that it can achieve extremely low MSE and MAE. The Transformer model has the second-best performance with highly satisfactory results in some countries, and the other models have poorer performance. This project highlights the high accuracy of the LSTM model, which can be used to predict the spread of COVID-19 so that countries can be better prepared and aware when controlling the spread.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

This module comprises framework plan of the venture with its architecture diagram, for example, generally design outline and cycle stream chart which tells about the modules mix in the task.

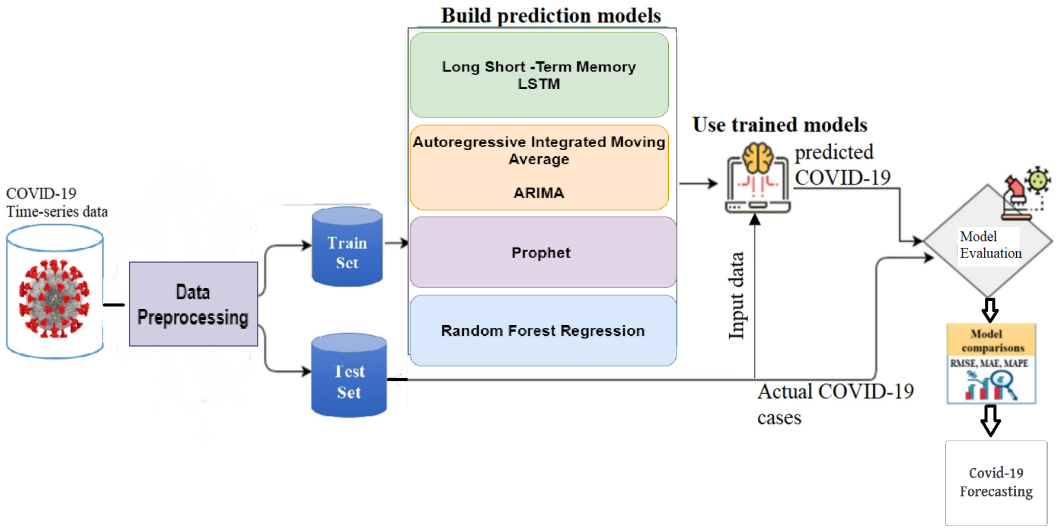


Figure 3.1: System Architecture

Figure 3.1 depicts the overall system architecture of the proposed system. For analysis the time series dataset will be taken and it will be pre-processed. That pre-processed data will be splitted into training and testing. The trained dataset will be applied to the LSTM model, Prophet model, ARIMA and Random Forest Regression model these models will be used for evaluating the test dataset. After the results the model will be used for future forecasting to forecast the upcoming cases and the evaluation of those results will be done in RMSE, MSE and MAE metrics.

3.2 MODULES DESCRIPTION

In this section I will explain about the list of modules which have been used in this project.

3.2.1 Data pre-processing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

3.2.2 Building Model

LSTM

I have implemented a simple Long Short-Term Memory (LSTM) model with an input layer, a single hidden layer, and an output layer that is used to make a prediction. The input layer has some neurons equal to 7 sequence steps (for one week COVID-19 data points). The hidden layer is an LSTM layer with 10 hidden units (neurons) and a rectified linear unit (ReLU) as an activation function. The output layer had a dense layer with 1 unit for predicting the output. The learning rate is set to 0.001, and it decays every five epochs. Moreover, we have used 1000 as the number of epochs, Adam as the optimizer, and the mean square error as the loss function. After that, we fit the model with prepared data to make a prediction. The obtained results may vary given the stochastic nature of the LSTM model; therefore, we have run it several times. Finally, we enter the last sequence with output to forecast the next value in the series.

PROPHET

I have extracted and collected COVID-19 cases related data from many sources and carried out pre-processing over the accumulated data to get good results with high precision while applying a model to fit the data. Generally the data we procure is sometimes in its most raw form i.e. irregular, imprecise and typically does not have particular characteristics. Feeding raw data to a model will cause the model to fail catastrophically (or) raise a good number of errors. This is where pre-processing gets in. Preprocessing helps to transform raw data to an organized and a tidy manner consequently making the data usable. The modelling we proposed is to obtain the prediction of the number of cases in the coming future. The data we dealt with is time-series, which means data that keeps changing with time, and our research matches this fact as COVID-19 cases keep changing with time. There are many models to fit time-series data, and we opted ”Prophet” as it is recognized to be accurate and quick and can handle outliers in data too. The working model of the prophet, mentioned in the above sections. In our research, we fit the prophet model on the existing data we have, which includes the dates and the number of cases on those days, respectively, and predicted the future cases for the next week, next month, and even the month after that. We plotted the time-series forecast curve to see how the cases will increase (or) decreased with time in the coming days and months.

ARIMA

I have acquired the data set for the ARIMA model from the WHO official COVID-19 site. The data was pulled from the official sources using a Web-Application and is updated daily. We used data from the date COVID-19 was first discovered in world on January 3, 2020, until May 18, 20222.Auto-Regressive Integrated Moving Average model, or ARIMA, is a

time-series forecasting approach that is used in predicting the future value of a variable from its own past values. It uses auto-regression and moving average, and incorporates a differencing order to remove trend and/or seasonality.

The ARIMA model contains 3 parameters (, ,). Parameter in the ARIMA model represents the periods to lag for. For instance, a value of = 2 indicates that 2 previous time periods of the time-series are used in the auto-regression part of the equation. Parameter represents the number of differencing transformations done to remove trend and/or seasonality therefore turning the time-series into a stationary one, i.e, making the mean and variance constant over time. This is an essential step to prepare the data for use in an ARIMA model. Parameter represents the lag of the error component of the ARIMA model. The error component is the part of the time-series that cannot be explained by trend or seasonality.

RANDOM FOREST

RF regression models are ensemble machine learning algorithms that were first described by Breiman. They createmultiple regression trees trained on unique bootstrap samples of the training dataset with a random subset of the inputfeatures. The output of all trees is averaged to create the final projection. We used the Scikit-learn (version 0.23.2)implementation with default hyperparameters and 20 estimators. The dataset used to train and validate the RF forecasting models includes Date of cases and number of cases on that day. however, “Weekly case increase” and “Daily estimated Rt” features were replaced. Thus, the preliminary forecasts were used as features to generate the final forecasts. All features were normalized by removing their mean and scaling to unit variance. For each Sunday from 11/01/2020 to 01/10/2021, I trained and validated RF forecasting models via incremental learning. Thus, I filtered out feature data that occurs on or after the Sunday of interest, but later

Sundays will have more feature data than earlier Sundays. Then, I randomly split the feature data for a given Sunday into a training subset and a validation subset with a 8:2 ratio, respectively. The training subset is used to train a RF regression model for each forecast of testing cases, the target outcome was the "daily case increase" to represent future cases on the forecast timepoint. The validation subset is used to ensure that trained RF models do not overfit the training subset. Finally, data for the Sunday of interest is input to the trained RF models to generate forecasts for each forecast timepoint.

3.2.3 Training Model

To train the model, we will take out training data 80 percentage and used 20 percentage of it as validation data. To higher the learning rate of our model we will use time series generator in the model. Training the model with n epochs.

3.2.4 Model Evaluation

It estimates how well (or how bad) the model is, in terms of its ability in mapping the relationship between X (a feature, or independent variable, or predictor variable) and Y (the target, or dependent variable, or response variable).

RMSE

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSD represents the square root of the second sample moment

of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors (or prediction errors) when computed out-of-sample. The RMSD serves to aggregate the magnitudes of the errors in predictions for various data points into a single measure of predictive power. RMSD is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent.

RMSD is always non-negative, and a value of 0 (almost never achieved in practice) would indicate a perfect fit to the data. In general, a lower RMSD is better than a higher one. However, comparisons across different types of data would be invalid because the measure is dependent on the scale of the numbers used.

RMSD is the square root of the average of squared errors. The effect of each error on RMSD is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSD. Consequently, RMSD is sensitive to outliers.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

Figure 3.2: Root Mean Squared Error

MSE

The Mean Squared Error (MSE) or Mean Squared Deviation (MSD) of an estimator measures the average of error squares i.e. the average squared difference between the estimated values and true value. It is a risk function, corresponding to the expected value of the squared error loss. It is always non – negative and values close to zero are better. The MSE is the second moment of the error (about the origin) and thus incorporates both the variance of the estimator and its bias.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2$$

Figure 3.3: Mean Squared Error

MAE

In statistics, mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as the sum of absolute errors divided by the sample size.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Figure 3.4: Mean Absolute Error

3.2.5 Visualization and Forecasting

By using the above models and analysing the evaluation metrics I can forecast the COVID-19 cases for the next year, And also to visualize the graph for the results I have used matplotlib and plotly library for a beautiful results.

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 HOME PAGE MODULE

The Figure 4.1 depicts the Home page screen provided. The purpose of this module is to upload csv format datasets which will be used for predictions and analysis.

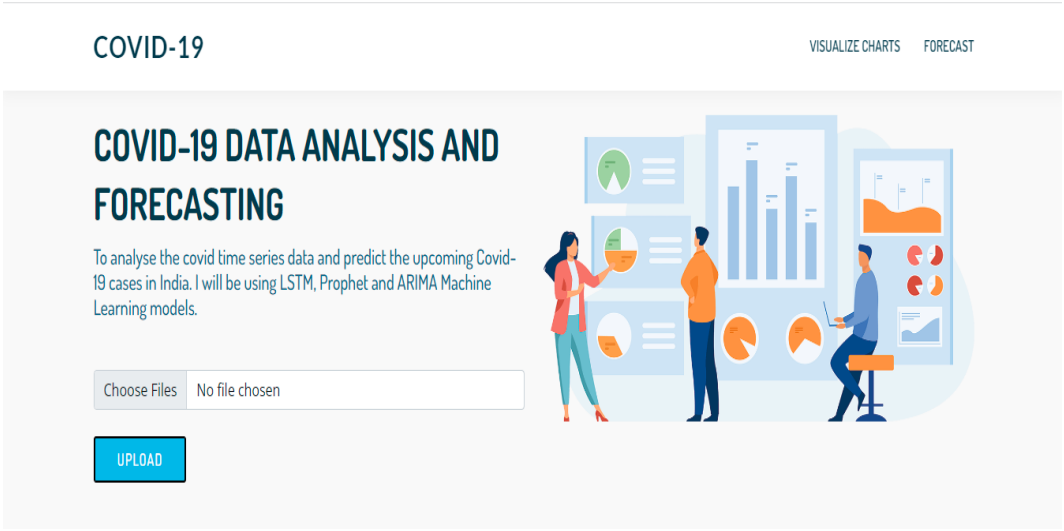


Figure 4.1: Home Screen

4.2 FILES MODULE

The Figure 4.2 depicts the files module which displays the list of files which have uploaded for analysis and prediction.

The files are specifically only should be in csv format, Which consumes less memory and read the file faster

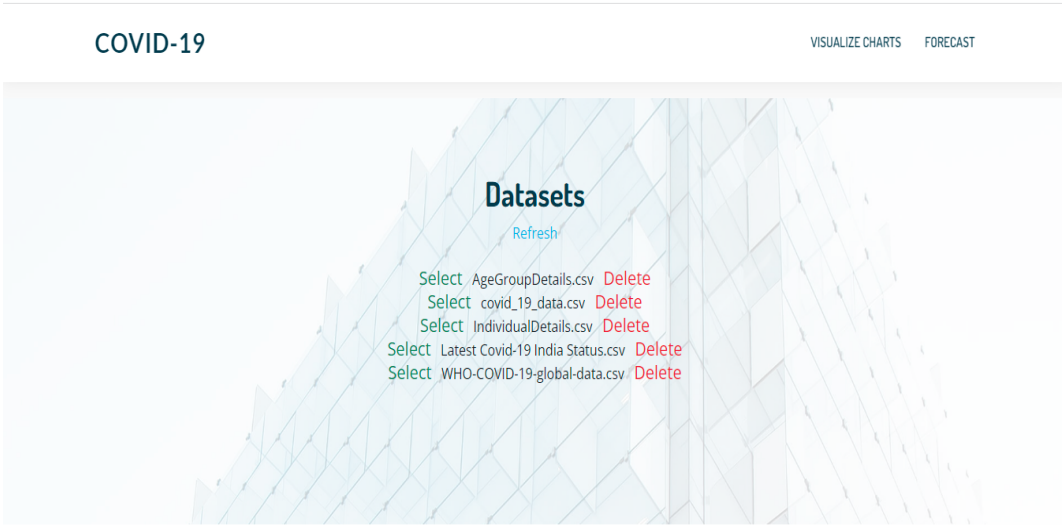


Figure 4.2: Files

4.3 MACHINE LEARNING MODELS

This module is designed to predict the covid-19 cases using the following machine learning models. The each model will read the csv data and split the data into training, testing and use the training data to train the model and then predict the cases for up coming days and also by comparing the results with the testing dataset. After the comparison I can able to get the evaluation metrics of the results in MAE, MSE, RMSE.

4.3.1 LSTM AND PROPHET MODEL

The Figure 4.3 depicts the first two models LSTM and Prophet are displayed.

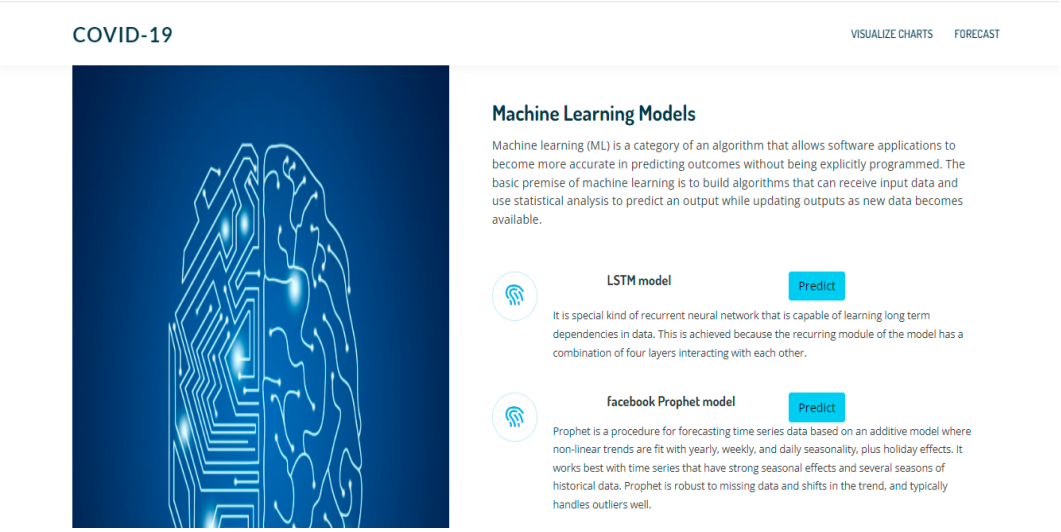


Figure 4.3: Machine Learning Models

4.3.2 ARIMA AND RANDOM FOREST MODEL

The Figure 4.4 depicts the next two models ARIMA and Random Forest are displayed.

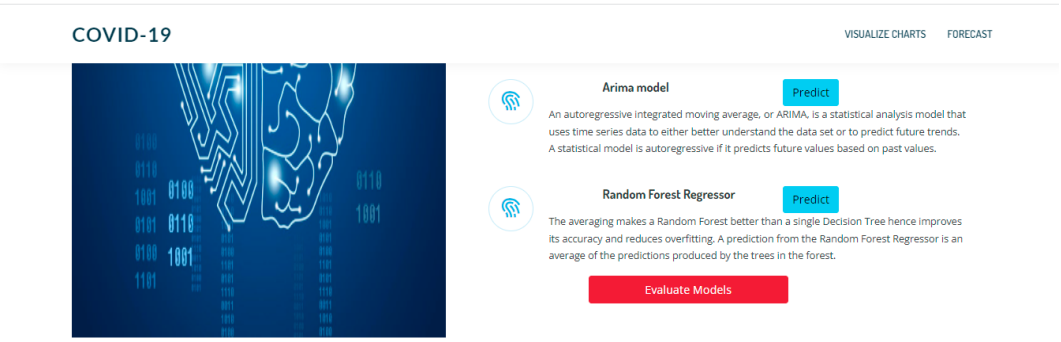


Figure 4.4: Machine Learning Models

4.4 EXPLORATORY DATA ANALYSIS

This module is designed to analyse the dataset and visualize the key

information from them.The Figure 4.5 displays the total population of states in India.

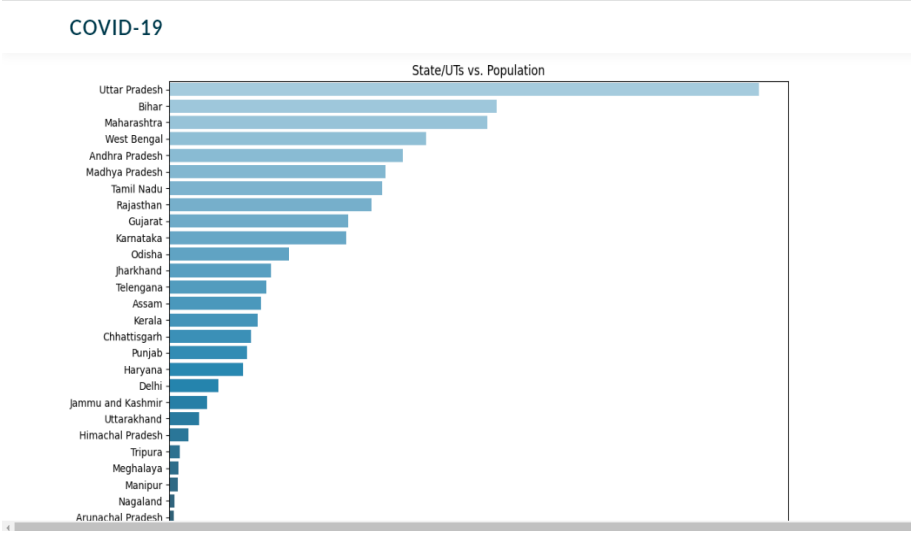


Figure 4.5: Population of states and UT

The Figure 4.6 displays the total cases in states of India.

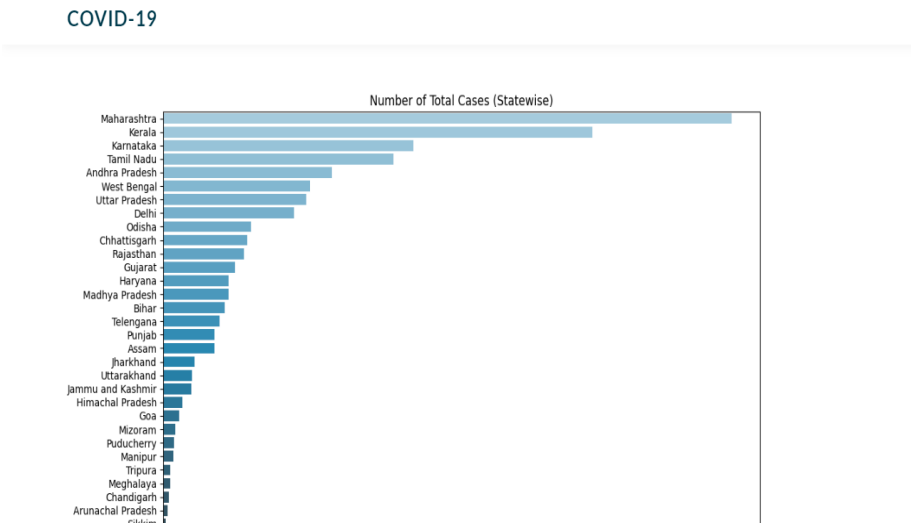


Figure 4.6: Maximum covid cases in states and UT

The Figure 4.7 displays the total active cases in states of India.

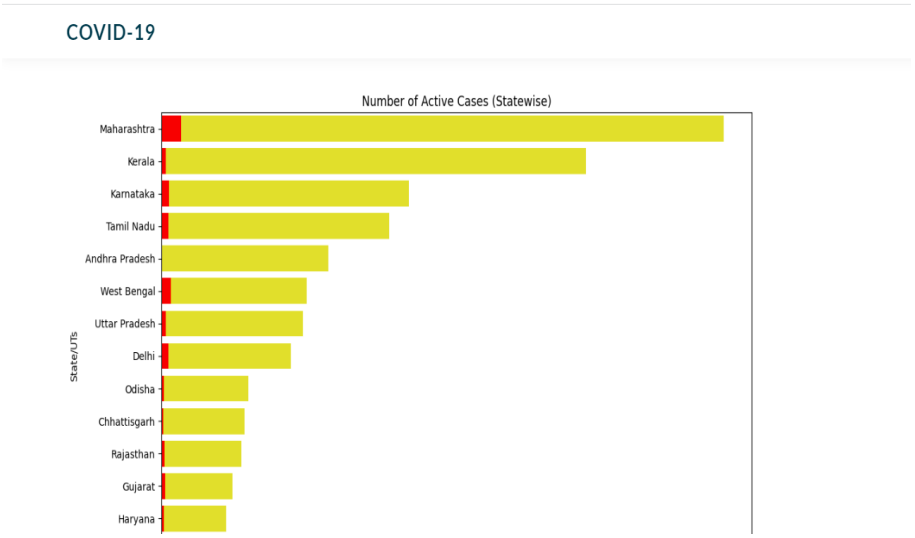


Figure 4.7: Active covid cases in states and UT

The Figure 4.8 displays the total deaths in states of India.

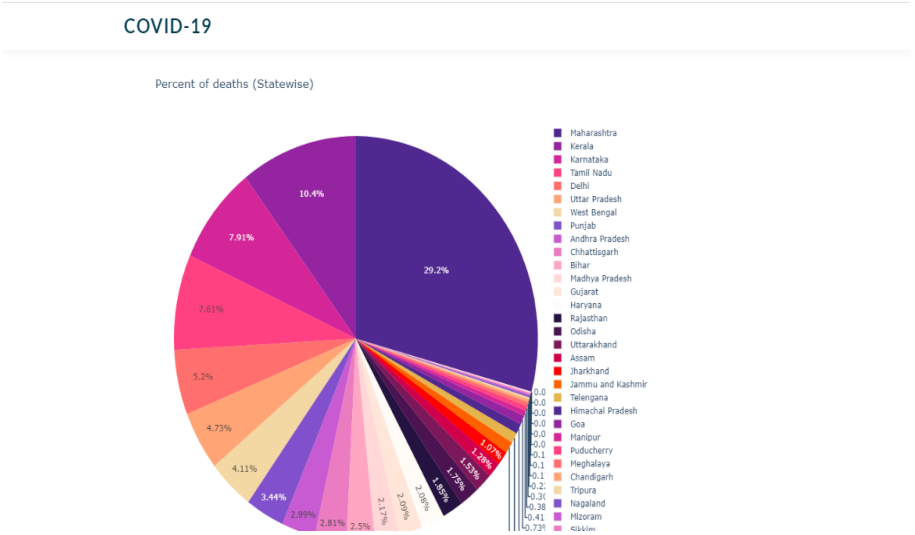


Figure 4.8: Total deaths of covid cases in states and UT

The Figure 4.9 displays the lockdown analysis of how it affected the increase/decrease of cases in states of India.

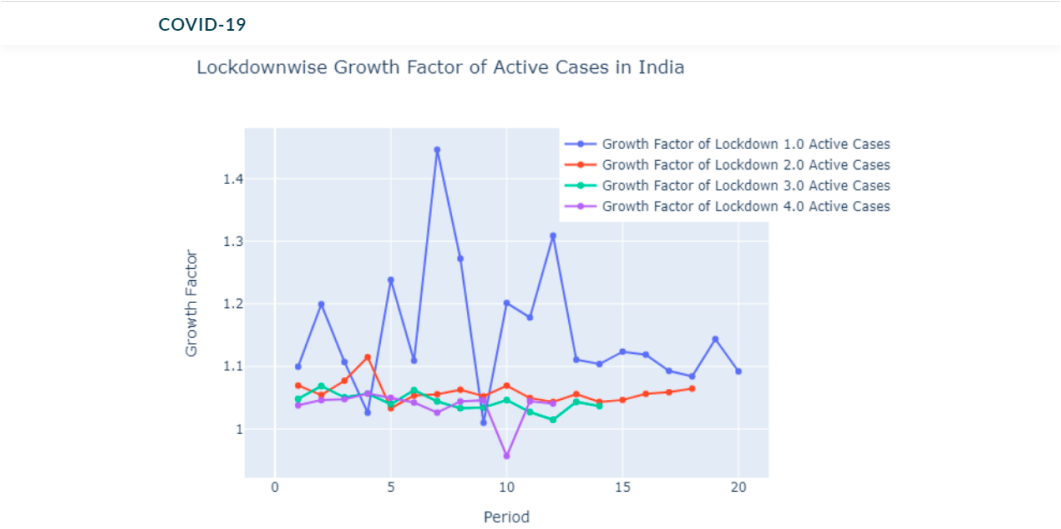


Figure 4.9: Lockdown analysis

The Figure 4.11 displays the covid cases of different age group of people.

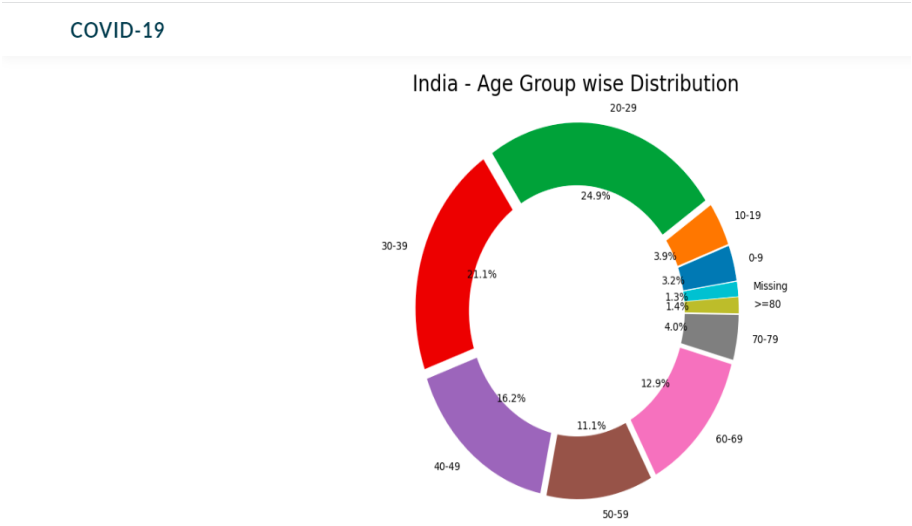


Figure 4.10: Cases based on age

The Figure 4.11 displays the covid cases based on gender.

COVID-19

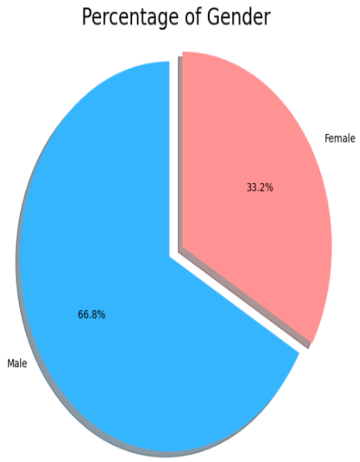


Figure 4.11: Cases based on gender

The Figure 4.12 displays the wave of covid cases from beginning.

COVID-19

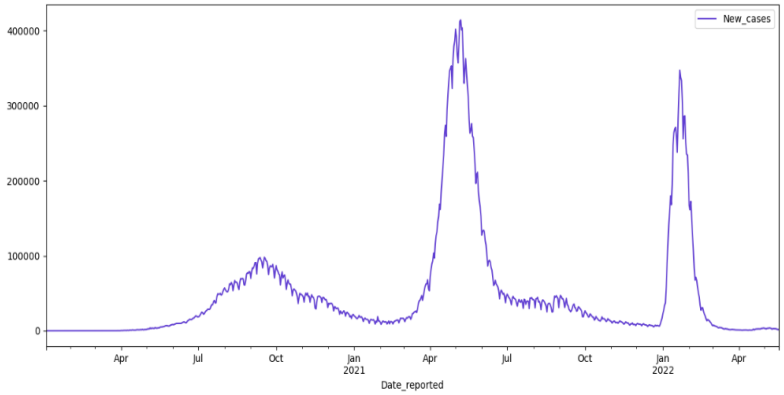
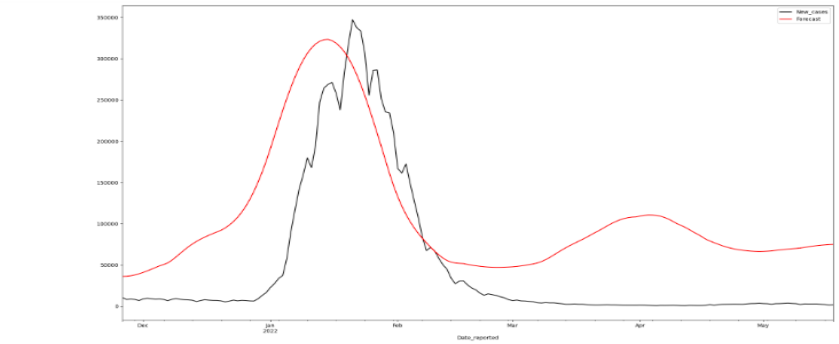


Figure 4.12: All cases

4.5 LSTM MODEL RESULTS

This module display how the LSTM model predicted the cases and how it is performed in the process with comparison of actual cases.



Mean Absolute Error (MAE) = 71269.10871074298

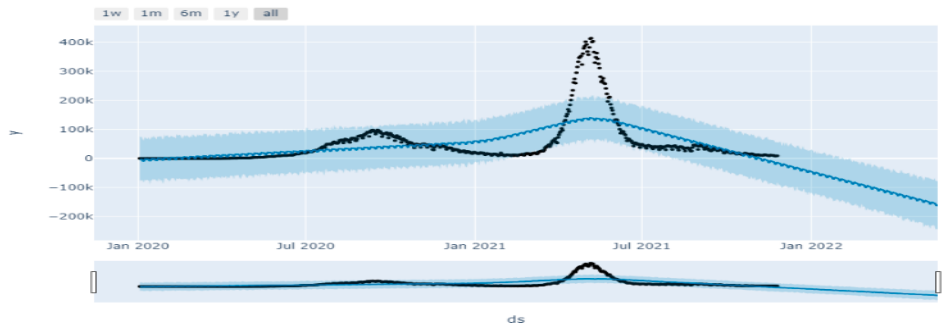
Mean Squared Error (MSE) = 6592136520.412533

Root Mean Squared Error (RMSE) = 81191.97325113199

Figure 4.13: LSTM Model Results

4.6 PROPHET MODEL RESULTS

This module display how the Prophet model predicted the cases and how it is performed in the process with comparison of actual cases.



Mean Absolute Error (MAE) = 55397.00280770074

Mean Squared Error (MSE) = 7929814506.702507

Root Mean Squared Error (RMSE) = 89049.50593182708

Figure 4.14: Prophet Model Results

4.7 ARIMA MODEL RESULTS

This module display how the ARIMA model predicted the cases and how it is performed in the process with comparison of actual cases.

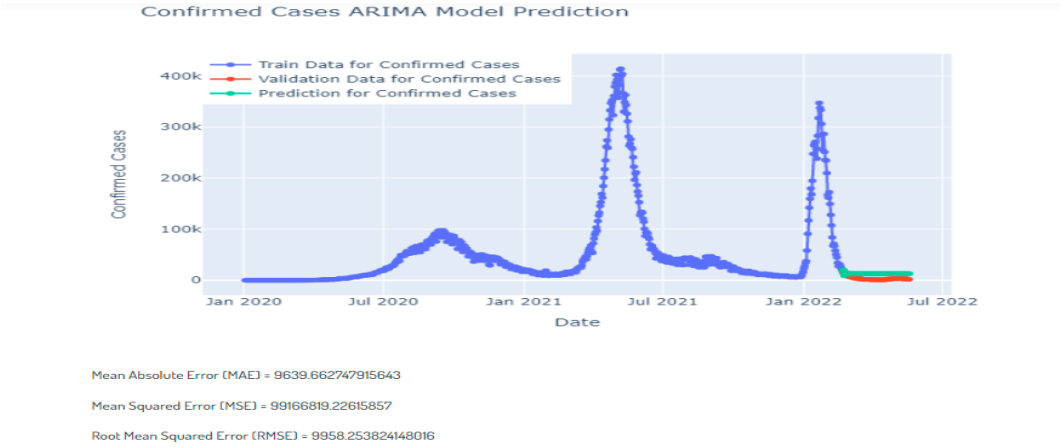


Figure 4.15: ARIMA Model Results

4.8 RANDOM FOREST MODEL RESULTS

This module display how the Random Forest model predicted the cases and how it is performed in the process with comparison of actual cases.

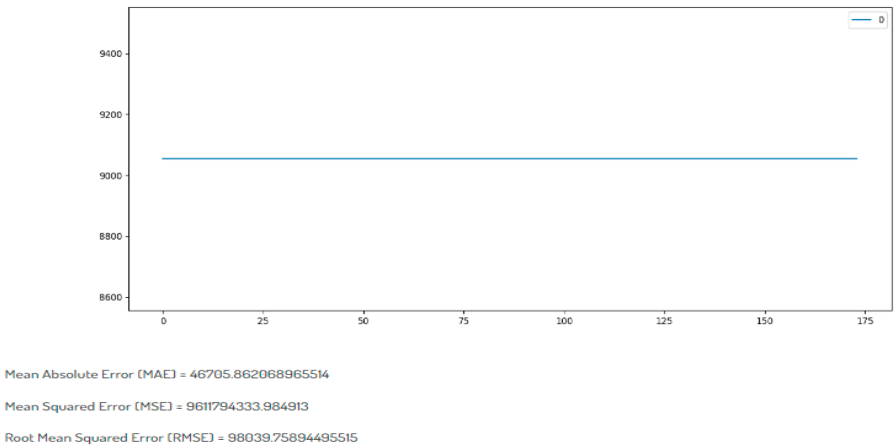


Figure 4.16: Random Forest Model Results

4.9 EVALUATION METRICS

The purpose of this module is to compare all the models and to choose the better model for future forecasting. I can see in 4.17 that ARIMA model has performed better as it has low RMSE.

	Model Name	Mean Squared Error	Mean Absolute Error	Root Mean Squared Error
2	ARIMA Model	99166819.226159	9639.662748	9958.253824
0	LSTM Model	6592136520.412533	71269.108711	81191.973251
1	Prophet Model	7929814506.702507	55397.002808	89049.505932
3	Random Model	46702.188659	9612964505.627542	98045.726606

Figure 4.17: Metrics

4.10 FUTURE FORECASTING

The Figure 4.18 shows the covid cases for the upcoming year in India.

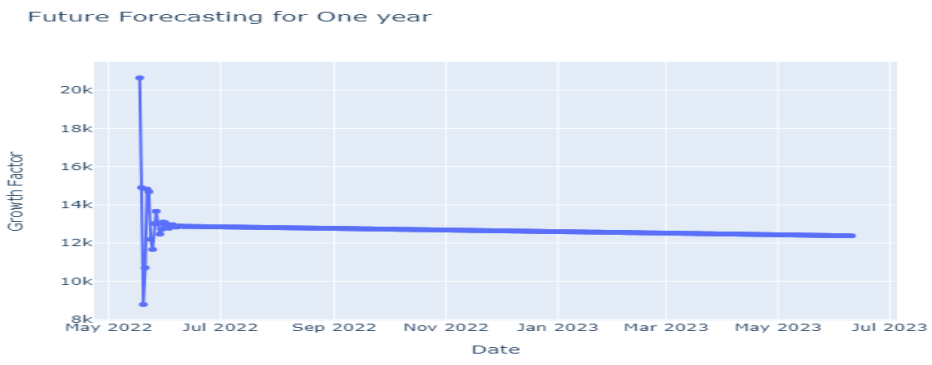


Figure 4.18: Forecasting

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

This project focused on applying machine-learning algorithms for predicting the COVID-19 cases for upcoming year. The results found on different machine-learning algorithms were able to predict that there will be a raise in COVID-19 cases for upcoming year. The ARIMA, LSTM, Prophet and Random Forest all of them showed promising results in the COVID-19 cases forecasting. My conclusion is ML algorithms in health showed promising results with high accuracy, sensitivity, and specificity using different models and algorithms.

5.2 FUTURE WORK

In this project, only four time-series analysis methods are used to predict the COVID-19 case trend and compare with each other. There are actually many more models that can be used in such time-series analysis scenario, such as linear regression and Convolutional Neural Network. In the future work, such models can be included to broaden the comparison. Only univariate data is considered in this project. Many researchers have proven that more sources of data can help increase the accuracy of prediction. In this case, the death and recovered case data, the vaccination rate, and other COVID-19 related data can help better model the disease trend.

REFERENCES

[1] Heamn N.Abduljabbar Ameer Sardar Kwekha Rashid and Bilal Alhayani. In *Coronavirus disease (COVID-19) cases analysis using machine-learning applications*, pages 501–513, 2020.

[2] C.K. AnitaYadav and Jha AditiSharan. In *Optimizing LSTM for time series prediction in Indian stock market*, pages 20–28, 2010.

[3] Tolga and Suleyman Serdar Kozat. Efficient online learning algorithms based on lstm neural networks. *IEEE transactions on nural networks and learnings ystems Journal*, 29(8):3772–3783, 2017.

[4] Sachin Aryal Ishan Manandhar Patricia B. MunroeAmeer Sardar Kwekha Rashid Ahmad Alimadadi. Artificial intelligence and machine learning to fight covid-19. In *AI and Machine Learning for Understanding Biological Processes*, pages 51–73, 2020.

[5] Tolga and Suleyman Serdar Kozat. Coronavirus disease 2019 (covid-19): current status and future perspectives. *International Journal of Antimicrobial Agents*, 29(8):257–289, 2020.